

Cross compile for Windows under Linux

This is much more convenient than building under windows, and the build is much faster. The resulting binary can be tested in wine. If you have any questions Cruisers Forum Thread is here: [building-opencpn-using-mingw-gcc-on-windows-121802.html#post1600285](https://cruisersforum.com/threads/building-opencpn-using-mingw-gcc-on-windows-121802.html#post1600285) . These instructions are less detailed, so it is helpful to first review and understand the build process for compiling using MinGW under windows.

Install mingw cross compiler, and wine for testing:

```
$ apt-get install mingw32 wine
```

cross compile wxwidgets:

```
$ ./configure --prefix=/usr/local/i586-mingw32 --host=i586-mingw32msvc --with-opengl --enable-unicode
$ make
$ sudo make install
```

note: you may attempt `--with-graphics_ctx` as well, but this requires linking with non-free `gdiplus.dll` and isn't required.

In `/usr/share/cmake-*/Modules/FindwxWidgets.cmake`

There is a bug in the cmake module `FindwxWidgets.cmake` that prevents any library using `wxWidgets` to cross compile for Windows from Linux. Please note that this bug is different from issue 0006187. Lines 192 to 198 are concerned by the present bug:

```
IF(WIN32 AND NOT CYGWIN AND NOT MSYS)
SET(wxWidgets_FIND_STYLE "win32")
ELSE(WIN32 AND NOT CYGWIN AND NOT MSYS)
IF(UNIX OR MSYS)
SET(wxWidgets_FIND_STYLE "unix")
ENDIF(UNIX OR MSYS)
ENDIF(WIN32 AND NOT CYGWIN AND NOT MSYS)
```

The problem is that, when cross compiling from Linux for Windows, the test `"WIN32 AND NOT CYGWIN AND NOT MSYS"` is true but find style should be `"unix"`, because the Windows-based `wxWidget` library is installed in a Unix environment.

Therefore, the following line replacement corrects this bug:

```
IF(WIN32 AND NOT CYGWIN AND NOT MSYS AND NOT CMAKE_CROSSCOMPILING)
SET(wxWidgets_FIND_STYLE "win32")
ELSE(WIN32 AND NOT CYGWIN AND NOT MSYS AND NOT CMAKE_CROSSCOMPILING)
```

```
IF(UNIX OR MSYS OR CMAKE_CROSSCOMPILING)
SET(wxWidgets_FIND_STYLE "unix")
ENDIF(UNIX OR MSYS OR CMAKE_CROSSCOMPILING)
ENDIF(WIN32 AND NOT CYGWIN AND NOT MSYS AND NOT CMAKE_CROSSCOMPILING)
```

After making this change, cross compile opencpn for windows.

```
$ git clone -b mingw https://github.com/seandepagnier/OpenCPN.git
$ cd OpenCPN
```

Copy the wxWidgets libraries into a folder here called wxWidgets-mingw

```
$ mkdir build-mingw32-cross
$ cd build-mingw32-cross
$ cmake -DCMAKE_TOOLCHAIN_FILE=../buildwin/Toolchain-mingw32.cmake ..
$ make
$ make package
$ ./opencpn*exe
```

Unfortunately “make install” does not work as it attempts to use the unix path. It may be possible to configure cmake iwth -DCMAKE_INSTALL_PREFIX= to allow it to work. I am not sure.

For make package, you must have nsis installed, I recommend version 3.00b. Running the package and installing opencpn copies everything into ~/.wine/drive_c/Program Files/OpenCPN

From there you can run opencpn without problems using wine. For building out of tree plugins, use exactly the same process.

From:
<https://opencpn.org/wiki/dokuwiki/> - **OpenCPN Manuals**

Permanent link:
https://opencpn.org/wiki/dokuwiki/doku.php?id=opencpn:developer_manual:developer_guide:compiling_windows:cross_compiling_for_windows_under_linux

Last update: 2017/02/23 14:04

